



PAT-NO: JP407045085A
DOCUMENT-IDENTIFIER: JP 07045085 A
TITLE: DATA WRITER AND READER
PUBN-DATE: February 14, 1995

INVENTOR-INFORMATION:

NAME	COUNTRY
HORI, FUSAO	

ASSIGNEE-INFORMATION:

NAME	COUNTRY
TEC CORP N/A	

APPL-NO: JP05189722
APPL-DATE: July 30, 1993

INT-CL (IPC): G11C016/06

ABSTRACT:

PURPOSE: To reduce the changing frequency of bit states at the time of program writing and to extend the service life of a rewritable ROM by counting the number of bits '0' and the number of bits '1', and when the number of bits '0' is less than the number of bits '1', writing data without inverting them.

CONSTITUTION: The number of bits '0' out of all program data obtained from data lines D0 to D7 connected from a data bus 15 to an EPROM 14 is counted, and when the number of bits '0' is smaller than a half of all bits constituting the EPROM 14, all program data are written in the EPROM 14 while holding the output of an inversion control circuit 17 at a low level. When the number of bits '0' is larger than a half of all the bits, the output of the circuit 17 is turned to a high

level and all the program data are inverted and written in the EPROM 14. Since the changing frequency of bits constituting the EPROM 14 from '1' to '0' can be reduced, stress can be reduced and the life of the EPROM 14 can be extended.

COPYRIGHT: (C)1995,JPO

* NOTICES *

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention writes data, such as a program, in ROM (read only memory) in which a rewrite is possible, for example, relates to the data write-in equipment and the data readers which use ROM in which the program was written by the ROM writer and this ROM writer, and in which a rewrite is possible, such as a system.

[0002]

[Description of the Prior Art] Although a rewrite is impossible for a general mask ROM etc., a memory content is eliminated electrically and ultraviolet rays, EPROM (erasable programmable read only memory) which can write in data again, EEPROM (electrically erasable programmable read only memory), and the flash memory are known also for nonvolatile memory. Such nonvolatile memory can be used as a ROM in which a rewrite is possible.

[0003] The ROM writer of the outline configuration shown in drawing 5 is used for ROM in which such a rewrite is possible conventionally, and data, such as a program, are written in.

[0004] EPROM1 as a ROM in which a rewrite is possible is connected to a data bus 2 through a connector. CPU (central processing unit)3 which constitutes a control-section body is connected with ROM4 and RAM (random access memory)5 through this data bus 2. The program data of the processing which said CPU3 performs are memorized by said ROM4, and the area of the various memory used for it when said CPU3 processes to said RAM5 is formed in it.

[0005] In such a ROM writer, the program data memorized by external storage, such as program data memorized by ROM4 or RAM5, a floppy disk, or a hard disk, are memorized by EPROM1 through a data bus 2 by control of CPU3.

[0006] By the way, as for ROM in which rewrites, such as EPROM, are possible, the count of writing has 100000 limits etc. When it changes the condition of the bit of each memory to 0 (low level) from 1 (high-level), or when changing it to 0-1, it is for stress to start the component which constitutes each bit of the above-mentioned rewrite possible ROM. And generally, ROM in which a rewrite is possible sets the condition (data) of all bits to 1, and is initialized.

[0007]

[Problem(s) to be Solved by the Invention] As mentioned above, the count of writing (life) is restricted by the stress which generates ROM in which a rewrite is possible when all bits change the bit which is initialized in the condition of 1 and constitutes ROM in which a rewrite is possible to the condition of 0 from the condition of 1.

[0008] Therefore, when writing in data, such as a program, it is possible to decrease stress and to prolong the life of ROM in which a rewrite is possible by lessening the number which changes a bit from the condition of 1 to the condition of 0.

[0009] Then, this invention aims at offering the data write-in equipment and the data reader which can lessen the number which changes a bit from the condition of 1 to the condition of 0, therefore can prolong the life of ROM in which a rewrite is possible, when writing in data, such as a program.

[0010]

[Means for Solving the Problem] In the data write-in equipment which writes data, such as a program, in the read only memory in which the rewrite of invention of claim 1 correspondence is possible counting which carries out counting of the number of bits of the condition of zero of all the data written in the read only memory in which a rewrite is possible, or the condition of 1 -- with a means this counting -- based on the number of bits by which counting was carried out with the means with a decision means to judge whether the number of bits of the condition of 0 in all data is larger than the number of bits of the condition of 1 With this decision means, when the number of bits of the condition of 0 in all data is below the number of bits of the condition of 1 The write-in means which reverses all data to the read only memory in which a rewrite is possible, and is written in it at the read only memory in which a rewrite is possible when the number of bits of writing and the condition of 0 is larger than the number of bits of the condition of 1 is established without reversing all data.

[0011] In the data reader which reads data, such as a program, in the read only memory in which a rewrite is possible, invention of claim 2 correspondence establishes a reversal reading means to reverse and read all those data in the read only memory in which a rewrite is possible, when the memory check result by memory check means to perform a memory check to the read only memory in which a rewrite is possible, and this memory check means becomes an error.

[0012]

[Function] invention of claim 1 correspondence -- setting -- counting -- it is judged whether the number of bits of the condition of 0 in all the data that counting of the number of bits of the condition of zero of all the data written in the read only memory in which a rewrite is possible with a means, or the condition of 1 is carried out, and it writes in the read only memory in which a rewrite is possible with a decision means based on this number of bits by which counting was carried out is larger than the number of bits of the condition of 1.

[0013] When it is not reversed and all data are written in the read only memory in which a rewrite is possible as it is, when the number of bits of the condition of 0 in all the data written in the read only memory in which a rewrite is possible by the decision is below the number of bits of the condition of 1, and the number of bits of the condition of 0 is judged to be larger than the number of bits of the condition of 1 by the write-in means, it is reversed and all data are written in the read only memory in which a rewrite is possible.

[0014] In invention of claim 2 correspondence, when the memory check of the read only memory in which a rewrite is possible is performed by the memory check means and the memory check result becomes an error, by the reversal reading means, all the data are reversed and are read in the read only memory in which a rewrite is possible.

[0015]

[Example] Hereafter, the example of this invention is explained with reference to a drawing.

[0016] Drawing 1 and drawing 2 show the example of invention (data write-in equipment) of claim 1 correspondence, and apply invention of claim 1 correspondence to a ROM (read only memory) writer.

[0017] Drawing 1 is the block diagram showing the important section circuitry of said ROM writer.

[0018] 11 is CPU (central processing unit) which constitutes a control-section body. RAM (random access memory) 13 in which the area of the various memory used when ROM12 the program data of the processing which this CPU11 performs were remembered to be, and said CPU11 process was formed, and EPROM (erasable programmable read only memory) 14 as read only memory in which a rewrite is possible are connected with said CPU11 through the data bus 15, the address bus (not shown), etc., respectively.

[0019] Said CPU11, said ROM12, and said RAM13 are connected with said data bus 15 by the eight data lines (D0-D7), respectively.

[0020] From the eight data lines (D0-D7) pulled out from said data bus 15, said EPROM14 is connected so that data may be inputted through exclusive OR circuits 160-167, respectively. The data out signal from the reversal control circuit 17 which while remains and becomes an input terminal from a flip-flop circuit of these exclusive OR circuits 160-167 is inputted as a common signal.

[0021] The data input terminal (D) of said reversal control circuit 17, the clock input terminal, and the reset input terminal of negative logic are connected with the IOWR-N output terminal (IOW) of the data line (D0) of said data bus 15, and the negative logic of said CPU11, and the reset signal line (RS), respectively. In addition, the reset signal line is connected also to the reset input terminal of the negative logic of said CPU11.

[0022] In addition, the data out signal from said reversal control circuit 17 serves as a low level by the pulse signal from the reset signal line (RS) inputted into this reversal control circuit 17, and the signal inputted into that data input terminal (D) is outputted by the standup of the signal inputted into that clock input terminal.

[0023] Therefore, unless said CPU11 starts from an IOWR-N output terminal (IOW) and outputs a signal, from said exclusive OR circuits 160-167, the data usually transmitted with the data bus 15 are inputted and written in said EPROM14 as it is.

[0024] The flow of the data write-in processing which said CPU11 performs to drawing 2 is shown.

[0025] First, address storage area N and number-of-bits storage area S formed in RAM13 is initialized to 0, counting of the number of bits of the condition of zero of the 1-byte data written in the address N stored in address storage area N of EPROM14 as processing of step 1 (ST1) is carried out (counting means), and this number of bits by which counting was carried out is stored in the temporary storage area T formed in RAM13.

[0026] Next, while adding number-of-bits T stored in number-of-bits S stored in number-of-bits storage area S in the temporary storage area T and storing in number-of-bits storage area S this number of bits added and obtained, addition processing of +1 is performed to the address N stored in address storage area N, and the address obtained by this addition processing is stored in address storage area N.

[0027] Next, it judges whether the address N stored in address storage area N is equal to last address L of EPROM14, and if the above-mentioned address N is not equal to the above-mentioned address L, it returns to processing of the above-mentioned step 1 again.

[0028] Moreover, if the above-mentioned address N is equal to the above-mentioned address L, it will judge whether it is larger than the number $(1/2) WL$ which number-of-bits S stored in the number-of-bits storage area carried out the multiplication of all the byte-count L of EPROM14 to number-of-bits W which constitutes 1-byte data, and did the division by 2 (decision means). That is, it judges whether the number of bits of the condition of 0 to write in is larger than the number of bits of the condition of 1 by whether the number of bits of the condition of zero of all the data written in EPROM14 is larger than the one half of the total number of bits which constitutes EPROM14.

[0029] Here, if the above-mentioned number-of-bits S is larger than the number WL of the above $(1/2)$, after it will output a high-level signal to the data line (D0) to the reversal control circuit 17, it starts from an IOWR-N output terminal (IOW) to the reversal control circuit 17, outputs a signal, makes the output of the reversal control circuit 17 high-level, reverses all the data written in EPROM14, and writes all the reversed data in EPROM14 (write-in means).

[0030] Moreover, if the above-mentioned number-of-bits S becomes below the number WL of the above $(1/2)$, all the data written in EPROM14 as it is will be written in EPROM14 (write-in means).

[0031] After the writing of all the data to EPROM14 is completed, this data write-in processing is ended.

[0032] In this example of such a configuration, the sum total of the number of bits of the condition of 0 is computed by crossing the program data memorized by external storage, such as ROM12, RAM13, or a floppy disk, to all the area (the address 0 - Address L) of EPROM14 about all the program data that counting of the number of bits of the condition (data) of 0 is carried out, and are written in for every address of every address of EPROM14, and the 1-byte data to write in. In below one half $(1/2) WL$ of the number of bits from which the sum total number of bits of this condition of computed 0 constitutes all the area of EPROM14, the output of the reversal control circuit 17 is left a low bell, and all program data are written in EPROM14.

[0033] Moreover, when the sum total number of bits of the condition of computed 0 is larger than the one half $(1/2) WL$ of the number of bits which constitutes all the area of EPROM14, the output of the

reversal control circuit 17 is made high-level, all program data are reversed, and it writes in EPROM14. [0034] Thus, the exclusive OR circuits 160-167 which were inserted in the eight data lines (D0-D7) connected to EPROM14 from the data bus 15 according to this example, As for each of these exclusive OR circuits 160-167, while remains, and the reversal control circuit 17 which outputs a signal to an input terminal in common is formed. In below one half of the total number of bits from which counting of the number of bits of the condition of zero of all the program data written in EPROM14 is carried out, and the number of bits of the condition of 0 constitutes EPROM14 All program data are written for the output of the reversal control circuit 17 in EPROM14 with a low level. When the number of bits of the condition of 0 is larger than the one half of the total number of bits which constitutes EPROM14 By making the output of the reversal control circuit 17 high-level, reversing all program data, and writing in EPROM14, the number which changes the bit which constitutes EPROM14 to the condition of 0 from the condition of 1 can be lessened. Therefore, the life of EPROM14 can be prolonged.

[0035] In addition, in this example, although what formed the reversal control circuit 17 which consists of a flip-flop circuit, and exclusive OR circuits 160-167 was explained, this invention is not limited to this, and when the number of bits of the condition (data) of zero is larger than the number of bits of the condition of 1, if the circuit and reversal process which reverse that data to write in are performed, it can be applied among the data to write in.

[0036] Drawing 3 and drawing 4 show the example of invention (data reader) of claim 2 correspondence, and apply it to the system which uses EPROM which wrote in the program for invention of claim 2 correspondence by the ROM writer of said 1st example.

[0037] Drawing 3 is the block diagram showing the important section circuitry of the read station of EPROM of said system.

[0038] 21 is CPU which constitutes a control-section body. RAM23 in which the area of the various memory used when ROM22 the program data of the processing which this CPU21 performs were remembered to be, and said CPU21 process was formed, and EPROM24 as read only memory in which a rewrite is possible are connected with said CPU21 through the data bus 25, the address bus (not shown), etc., respectively.

[0039] Said CPU21, said ROM22, and said RAM23 are connected with said data bus 25 by the eight data lines (D0-D7), respectively.

[0040] Said EPROM24 is connected so that data may be outputted to the eight data lines (D0-D7) pulled out from said data bus 25 through exclusive OR circuits 260-267, respectively. The data out signal from the reversal control circuit 27 which while remains and becomes an input terminal from a flip-flop circuit of these exclusive OR circuits 260-267 is inputted as a common signal.

[0041] The data input terminal (D) of said reversal control circuit 27, the clock input terminal, and the reset input terminal of negative logic are connected with the IOWR-N output terminal (IOW) of the data line (D0) of said data bus 25, and the negative logic of said CPU21, and the reset signal line (RS), respectively. In addition, the reset signal line is connected also to the reset input terminal of the negative logic of said CPU21.

[0042] Therefore, unless said CPU21 starts from an IOWR-N output terminal (IOW) and outputs a signal, from said exclusive OR circuits 260-267, the data usually read in EPROM24 are outputted to said data bus 25 as it is.

[0043] The flow of the data reading processing which said CPU21 performs to drawing 4 is shown.

[0044] First, address storage area N and number-of-bits storage area S formed in RAM23 is initialized to 0, counting of the number of bits of the condition of zero of the 1-byte data currently written in the address N stored in the address storage area of EPROM24 as processing of step 2 (ST2) is carried out, and these enumerated data are stored in the temporary storage area T formed in RAM23.

[0045] Next, while adding number-of-bits T stored in number-of-bits S stored in number-of-bits storage area S in the temporary storage area T and storing in number-of-bits storage area S this number of bits added and obtained, addition processing of +1 is performed to the address N stored in address storage area N, and the address obtained by this addition processing is stored in address storage area N.

[0046] Next, it judges whether the address N stored in address storage area N is equal to the address L-1

in front of [of last address L of EPROM24] one, and if the above-mentioned address N is not equal to the above-mentioned address L-1, it returns to processing of the above-mentioned step 2 again.

[0047] Moreover, if the above-mentioned address N is equal to the above-mentioned address L-1, the numeric value S+1 set to number-of-bits S stored in the number-of-bits storage area +one will judge whether it was in agreement with the contents (checksum value) of the 1-byte data currently written in the address L of EPROM24 (memory check means). That is, the checksum data written in the address L of EPROM24 perform a memory check.

[0048] Here, if the above-mentioned numeric value S+1 is in agreement with the checksum value of the above-mentioned address L, it will leave the output of the reversal control circuit 27 a low level, and will read all data in EPROM24.

[0049] Moreover, if the above-mentioned numeric value S+1 is not equal to the checksum value of the above-mentioned address L, after it judges that reversal data are written in EPROM24 and outputs a high-level signal to the data line (D0) to the reversal control circuit 27, it starts from an IOWR-N output terminal (IOW) to the reversal control circuit 27, a signal is outputted, the output of the reversal control circuit 27 is made high-level, and all data are reversed and read in EPROM24 (reversal reading means).

[0050] After the read of all the data from EPROM24 mentioned above is completed, this data reading processing is ended.

[0051] In this example of such a configuration, for every address of every address of EPROM24, and the 1-byte data to read, counting of the number of bits of the condition (data) of 0 is carried out, and, as for the program data memorized by EPROM24, the sum total of the number of bits of the condition of 0 to the data of the address L-1 of EPROM24 is computed.

[0052] Data for a checksum to collate are computed by furthermore adding to the sum total number of bits of the condition of 0 to this address L-1 +one time. The positive value data of a checksum are memorized to the address L of EPROM24, and judge whether the data (numeric value) added to the sum total number of bits mentioned above +one time as a memory check and the data (numeric value) memorized to the address L of EPROM24 were in agreement.

[0053] When it is judged that it was in agreement by this decision, the output of the reversal control circuit 27 is left a low level, and program data are read in EPROM24.

[0054] Moreover, by the decision, when it is judged that it is not in agreement (i.e., when judged as an error by the memory check), the output of the reversal control circuit 27 is made high-level, and program data are reversed and read in EPROM24.

[0055] Thus, according to this example, the reversal control circuit 27 which outputs a signal to **** one [each] input terminal of exclusive OR circuits 260-267 inserted in the eight data lines (D0-D7) connected to the data bus 25 from EPROM24 in common is formed. When the result of a checksum is judged to be normal When program data are read in EPROM24 and the result of a checksum is judged to be an error with a low level, the output of the reversal control circuit 27 By making the output of the reversal control circuit 27 high-level, and reversing and reading program data in EPROM24 Program data can be correctly read in ROM (EPROM) which wrote in data by the ROM writer of the 1st above-mentioned example and in which a rewrite is possible. Therefore, the life of EPROM24 can be prolonged.

[0056] In addition, although what formed the reversal control circuit 27 which consists of a flip-flop circuit, and exclusive OR circuits 260-267 in this example was explained, this invention can be applied, if the circuit or reversal process which reverses the data to read is performed, when it is not limited to this and judged as an error by the memory check.

[0057] Moreover, in two examples mentioned above, although EPROMs 14 and 24 were explained as a ROM in which a rewrite is possible, this invention is not limited to this and is applicable also about EEPROM (electrically erasable programmable read only memory) or a flash memory.

[0058]

[Effect of the Invention] As explained in full detail above, when writing in data, such as a program, according to this invention, the data write-in equipment and the data reader which can lessen the number which changes a bit from the condition of 1 to the condition of 0, therefore can prolong the life of ROM

JP,07-045085,A [DETAILED DESCRIPTION]

in which a rewrite is possible can be offered.

[Translation done.]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平7-45085

(43)公開日 平成7年(1995)2月14日

(51)Int.Cl.*	識別記号	庁内整理番号	F I	技術表示箇所
G 1 1 C 16/06		6866-5L	G 1 1 C 17/ 00	5 1 0 Z 3 0 9 F

審査請求 未請求 請求項の数2 O L (全 7 頁)

(21)出願番号 特願平5-189722
(22)出願日 平成5年(1993)7月30日

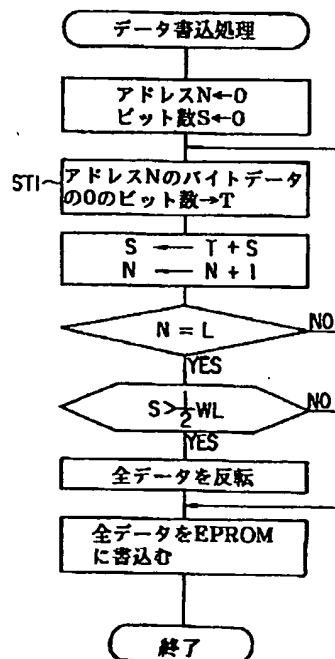
(71)出願人 000003562
株式会社テック
静岡県田方郡大仁町大仁570番地
(72)発明者 保里 房生
静岡県田方郡大仁町大仁570番地 東京電
気株式会社大仁工場内
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 データ書込装置及びデータ読取装置

(57)【要約】

【目的】プログラム等を書込時に、ビットの状態を1から0に変化させる数を少なくして、再書込み可能なROMの寿命を延ばす。

【構成】データバスからEPROMへのデータ線に介挿されたエクスクルーシブOR回路と、各エクスクルーシブOR回路の入力端子へ共通に接続された反転制御回路とを設け、全データのうち0のビット数が1のビット数より大きい場合に、反転制御回路の出力をハイレベルにして全データを反転させてEPROMに書込む書込装置及び、EPROMからデータバスへのデータ線に介挿されたエクスクルーシブOR回路と、各エクスクルーシブOR回路の入力端子へ共通に接続された反転制御回路とを設け、チェックサムエラー時に、反転制御回路の出力をハイレベルにしてEPROMからデータを反転して読取る読取装置。



【特許請求の範囲】

【請求項1】 再書き込み可能なリード・オンリ・メモリへプログラム等のデータを書込むデータ書込装置において、前記再書き込み可能なリード・オンリ・メモリに書込む全データのうちの0の状態又は1の状態のビット数を計数する計数手段と、この計数手段により計数されたビット数に基づいて、前記全データにおける0の状態のビット数が1の状態のビット数より大きいかな否かを判断する判断手段と、この判断手段により前記全データにおける0の状態のビット数が1の状態のビット数以下のときは、前記全データを反転せずに前記再書き込み可能なリード・オンリ・メモリに書き込み、0の状態のビット数が1の状態のビット数より大きいときには、前記全データを反転して前記再書き込み可能なリード・オンリ・メモリに書き込む書込手段とを設けたことを特徴とするデータ書込装置。

【請求項2】 再書き込み可能なリード・オンリ・メモリからプログラム等のデータを読取るデータ読取装置において、前記再書き込み可能なリード・オンリ・メモリに対してメモリチェックを行うメモリチェック手段と、このメモリチェック手段によるメモリチェック結果がエラーとなったときには、前記再書き込み可能なリード・オンリ・メモリからその全データを反転して読取る反転読取手段とを設けたことを特徴とするデータ読取装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、再書き込み可能なROM (read only memory) にプログラム等のデータを書込む、例えばROMライター及びこのROMライターでプログラムが書込まれた再書き込み可能なROMを使用するシステム等のデータ書込装置及びデータ読取装置に関する。

【0002】

【従来の技術】一般的なマスクROM等は再書き込みが不可能であるが、不揮発性メモリでも、紫外線や電気的にメモリ内容を消去して、再びデータを書込むことができるEPROM (erasable programmable read only memory) やEEPROM (electrically erasable programmable read only memory) 及びフラッシュメモリが知られている。このような不揮発性メモリは、再書き込み可能なROMとして使用することができる。

【0003】従来、このような再書き込み可能なROMは、例えば図5に示す概略構成のROMライターを使用して、プログラム等のデータが書込まれるようになっていた。

【0004】再書き込み可能なROMとしてのEPROM 1は、コネクタを介してデータバス2に接続される。このデータバス2を介して、制御部本体を構成するCPU (central processing unit) 3は、ROM4及びRAM (random access memory) 5と接続されている。前記ROM4には、前記CPU3が行う処理のプログラムデ

ータが記憶され、前記RAM5には、前記CPU3が処理を行う時に使用する各種メモリのエリアが形成されている。

【0005】このようなROMライターでは、CPU3の制御により、ROM4又はRAM5に記憶されたプログラムデータ、あるいはフロッピーディスク又はハードディスク等の外部記憶装置に記憶されたプログラムデータが、データバス2を介してEPROM1に記憶されるようになっている。

【0006】ところで、EPROM等の再書き込み可能なROMは、書き込み回数が、例えば100000回などの制限がある。それは、各メモリのビットの状態を1 (ハイレベル) から0 (ローレベル) へ変化させるときに、又は0から1へ変化させるときに、上記再書き込み可能なROMの各ビットを構成する素子にストレスがかかるためである。そして、一般的に再書き込み可能なROMは、全てのビットの状態 (データ) を1にして初期化される。

【0007】

【発明が解決しようとする課題】上述したように、再書き込み可能なROMは、全てのビットが1の状態に初期化され、再書き込み可能なROMを構成するビットを1の状態から0の状態に変化させるときに発生するストレスにより、書き込み回数 (寿命) が制限される。

【0008】従って、プログラム等のデータを書込むときに、ビットを1の状態から0の状態に変化させる数を少なくすることにより、ストレスを減少させて再書き込み可能なROMの寿命を延ばすことが考えられる。

【0009】そこでこの発明は、プログラム等のデータを書込むときに、ビットを1の状態から0の状態に変化させる数を少なくすることができ、従って再書き込み可能なROMの寿命を延ばすことができるデータ書込装置及びデータ読取装置を提供することを目的とする。

【0010】

【課題を解決するための手段】請求項1対応の発明は、再書き込み可能なリード・オンリ・メモリへプログラム等のデータを書込むデータ書込装置において、再書き込み可能なリード・オンリ・メモリに書込む全データのうちの0の状態又は1の状態のビット数を計数する計数手段と、この計数手段により計数されたビット数に基づいて、全データにおける0の状態のビット数が1の状態のビット数より大きいかな否かを判断する判断手段と、この判断手段により全データにおける0の状態のビット数が1の状態のビット数以下のときには、全データを反転せずに再書き込み可能なリード・オンリ・メモリに書き込み、0の状態のビット数が1の状態のビット数より大きいときには、全データを反転して再書き込み可能なリード・オンリ・メモリに書き込む書込手段とを設けたものである。

【0011】請求項2対応の発明は、再書き込み可能なリード・オンリ・メモリからプログラム等のデータを読取るデータ読取装置において、再書き込み可能なリード・オ

3

ンリ・メモリに対してメモリチェックを行うメモリチェック手段と、このメモリチェック手段によるメモリチェック結果がエラーとなったときには、再書き込み可能なリード・オンリ・メモリからその全データを反転して読取る反転読取手段とを設けたものである。

【0012】

【作用】請求項1対応の発明においては、計数手段により、再書き込み可能なリード・オンリ・メモリに書込む全データのうちの0の状態又は1の状態のビット数が計数され、判断手段により、この計数されたビット数に基づいて、再書き込み可能なリード・オンリ・メモリに書込む全データにおける0の状態のビット数が1の状態のビット数より大きいかが判断される。

【0013】書き込み手段により、その判断で再書き込み可能なリード・オンリ・メモリに書込む全データにおける0の状態のビット数が1の状態のビット数以下のときには、全データは、反転されずそのまま再書き込み可能なリード・オンリ・メモリに書込まれ、0の状態のビット数が1の状態のビット数より大きいと判断されたときには、全データは、反転されて再書き込み可能なリード・オンリ・メモリに書込まれる。

【0014】請求項2対応の発明においては、メモリチェック手段により、再書き込み可能なリード・オンリ・メモリのメモリチェックが行われ、そのメモリチェック結果がエラーとなったときには、反転読取手段により、再書き込み可能なリード・オンリ・メモリからその全データが、反転して読取られる。

【0015】

【実施例】以下、この発明の実施例を図面を参照して説明する。

【0016】図1及び図2は、請求項1対応の発明（データ書込装置）の実施例を示すもので、請求項1対応の発明をROM（read only memory）ライタに適用したものである。

【0017】図1は、前記ROMライタの要部回路構成を示すブロック図である。

【0018】11は、制御部本体を構成するCPU（central processing unit）である。このCPU11が行う処理のプログラムデータが記憶されたROM12、前記CPU11が処理を行うときに使用する各種メモリのエリアが形成されたRAM（random access memory）13及び再書き込み可能なリード・オンリ・メモリとしてのEPROM（erasable programmable read only memory）14は、それぞれデータバス15及びアドレスバス（図示せず）等を介して前記CPU11と接続されている。

【0019】前記CPU11、前記ROM12及び前記RAM13は、それぞれ前記データバス15と8本のデータ線（D0～D7）により接続されている。

【0020】前記EPROM14は、前記データバス1

4

5から引き出された8本のデータ線（D0～D7）から、それぞれエクスクルーシブOR回路160～167を介してデータを入力するように接続されている。このエクスクルーシブOR回路160～167の残る一方の入力端子には、フリップフロップ回路からなる反転制御回路17からのデータ出力信号が共通信号として入力されている。

【0021】前記反転制御回路17のデータ入力端子（D）、クロック入力端子及び負論理のリセット入力端子は、それぞれ前記データバス15のデータ線（D0）、前記CPU11の負論理のIOWR-N出力端子（IOW）及びリセット信号線（RS）と接続されている。なおリセット信号線は、前記CPU11の負論理のリセット入力端子にも接続されている。

【0022】なお、前記反転制御回路17からのデータ出力信号は、この反転制御回路17に入力されるリセット信号線（RS）からのパルス信号により、ローレベルとなり、そのクロック入力端子に入力される信号の立ち上がりにより、そのデータ入力端子（D）に入力されている信号が出力されるようになっている。

【0023】従って、前記CPU11がIOWR-N出力端子（IOW）から立上がり信号を出力しない限り、前記エクスクルーシブOR回路160～167からは、通常データバス15で転送されてきたデータが、そのまま前記EPROM14に入力され書込まれる。

【0024】図2に、前記CPU11が行うデータ書込処理の流れを示す。

【0025】まず、RAM13に形成されたアドレス格納エリアN及びビット数格納エリアSを0に初期化して、ステップ1（ST1）の処理として、EPROM14のアドレス格納エリアNに格納されたアドレスNに書込む1バイトデータのうちの0の状態のビット数を計数し（計数手段）、この計数されたビット数をRAM13に形成された一時格納エリアTに格納する。

【0026】次に、ビット数格納エリアSに格納されたビット数Sに一時格納エリアTに格納されたビット数Tを加算し、この加算して得たビット数をビット数格納エリアSに格納すると共に、アドレス格納エリアNに格納されたアドレスNに対して+1の加算処理を行い、この加算処理により得たアドレスをアドレス格納エリアNに格納する。

【0027】次に、アドレス格納エリアNに格納されたアドレスNが、EPROM14の最終アドレスLに等しいかを判断し、上記アドレスNが上記アドレスLに等しくなければ、再び前述のステップ1の処理に戻るようになっている。

【0028】また、上記アドレスNが上記アドレスLに等しいならば、ビット数格納エリアに格納されたビット数Sが、1バイトデータを構成するビット数WにEPROM14の全バイト数Lを乗算して2で除算した数（1

／2) WLより大きいかなかを判断する(判断手段)。
すなわち、EPROM14に書込む全データのうちの0
の状態のビット数が、EPROM14を構成する全ビッ
ト数の半分より大きいかなかに、書込む0の状態の
ビット数が1の状態のビット数より大きいかなかを判断
する。

【0029】ここで、上記ビット数Sが、上記数(1/
2) WLより大きいならば、反転制御回路17へのデー
タ線(D0)にハイレベルの信号を出力したのち、IOWR-N出力端子(IOW)から反転制御回路17へ立
上がり信号を出力し、反転制御回路17の出力をハイレ
ベルにして、EPROM14に書込む全データを反転
し、その反転した全データをEPROM14に書込む
(書込手段)。

【0030】また、上記ビット数Sが上記数(1/2)
WL以下ならば、そのままEPROM14に書込む全デ
ータをEPROM14に書込む(書込手段)。

【0031】EPROM14への全データの書込みが終
了すると、このデータ書込処理を終了するようになって
いる。

【0032】このような構成の本実施例においては、R
OM12又はRAM13あるいはフロッピーディスク等
の外部記憶装置に記憶されたプログラムデータは、EP
ROM14の各アドレス毎に、すなわち書込む1バイト
データのアドレス毎に、0の状態(データ)のビット数
が計数され、書込む全プログラムデータについて、EP
ROM14の全エリア(アドレス0～アドレスL)にわ
たって、0の状態のビット数の合計が算出される。この
算出された0の状態の合計ビット数が、EPROM14
の全エリアを構成するビット数の半分(1/2) WL以
下の場合には、反転制御回路17の出力をローレベルのま
まにして、全プログラムデータをEPROM14に書込
む。

【0033】また、算出された0の状態の合計ビット数
が、EPROM14の全エリアを構成するビット数の半
分(1/2) WLより大きい場合には、反転制御回路1
7の出力をハイレベルにし、全プログラムデータを反転
してEPROM14に書込む。

【0034】このように本実施例によれば、データバス
15からEPROM14へ接続された8本のデータ線
(D0～D7)に介挿されたエクスクルーシブOR回路
160～167と、この各エクスクルーシブOR回路1
60～167の残る一方の入力端子へ共通に信号を出力
する反転制御回路17とを設け、EPROM14に書込
む全プログラムデータのうちの0の状態のビット数を計
数し、0の状態のビット数がEPROM14を構成する
全ビット数の半分以下の場合には、反転制御回路17の
出力をローレベルのまま全プログラムデータをEPROM
14に書込み、0の状態のビット数がEPROM14
を構成する全ビット数の半分より大きい場合には、反転

制御回路17の出力をハイレベルにして全プログラムデ
ータを反転させてEPROM14に書込むことにより、
EPROM14を構成するビットを1の状態から0の状
態に変化させる数を少なくすることができる。従って、
EPROM14の寿命を延ばすことができる。

【0035】なお、この実施例においては、フリップフ
ロップ回路からなる反転制御回路17とエクスクルーシ
ブOR回路160～167とを設けたものについて説明
したが、この発明はこれに限定されるものではなく、書
込むデータのうち0の状態(データ)のビット数が1の
状態のビット数より大きいときに、その書込むデータを
反転する回路及び反転処理を行うものならば適用できる
ものである。

【0036】図3及び図4は、請求項2対応の発明(デ
ータ読取装置)の実施例を示すもので、請求項2対応の
発明を前記第1の実施例のROMライタでプログラムを
書込んだEPROMを使用するシステムに適用したもの
である。

【0037】図3は、前記システムのEPROMの読取
部の要部回路構成を示すブロック図である。

【0038】21は、制御部本体を構成するCPUであ
る。このCPU21が行う処理のプログラムデータが記
憶されたROM22、前記CPU21が処理を行うとき
に使用する各種メモリのエリアが形成されたRAM23
及び再書込み可能なリード・オンリ・メモリとしてのE
PROM24は、それぞれデータバス25及びアドレス
バス(図示せず)等を介して前記CPU21と接続され
ている。

【0039】前記CPU21、前記ROM22及び前記
RAM23は、それぞれ前記データバス25と8本のデ
ータ線(D0～D7)により接続されている。

【0040】前記EPROM24は、前記データバス2
5から引き出された8本のデータ線(D0～D7)へ、
それぞれエクスクルーシブOR回路260～267を介
してデータを出力するように接続されている。このエク
スクルーシブOR回路260～267の残る一方の入力
端子には、フリップフロップ回路からなる反転制御回路
27からのデータ出力信号が共通信号として入力されて
いる。

【0041】前記反転制御回路27のデータ入力端子
(D)、クロック入力端子及び負論理のリセット入力端
子は、それぞれ前記データバス25のデータ線(D
0)、前記CPU21の負論理のIOWR-N出力端子
(IOW)及びリセット信号線(RS)と接続されてい
る。なおリセット信号線は、前記CPU21の負論理の
リセット入力端子にも接続されている。

【0042】従って、前記CPU21がIOWR-N出
力端子(IOW)から立上がり信号を出力しない限り、
前記エクスクルーシブOR回路260～267からは、
通常EPROM24から読取られたデータが、そのま

7

ま、前記データバス25に出力される。

【0043】図4に、前記CPU21が行うデータ読取処理の流れを示す。

【0044】まず、RAM23に形成されたアドレス格納エリアN及びビット数格納エリアSを0に初期化して、ステップ2(ST2)の処理として、EPROM24のアドレス格納エリアに格納されたアドレスNに書込まれている1バイトデータのうちの0の状態のビット数を計数し、この計数値をRAM23に形成された一時格納エリアTに格納する。

【0045】次に、ビット数格納エリアSに格納されたビット数Sに一時格納エリアTに格納されたビット数Tを加算し、この加算して得たビット数をビット数格納エリアSに格納すると共に、アドレス格納エリアNに格納されたアドレスNに対して+1の加算処理を行い、この加算処理により得たアドレスをアドレス格納エリアNに格納する。

【0046】次に、アドレス格納エリアNに格納されたアドレスNが、EPROM24の最終アドレスLの1つ前のアドレスL-1に等しいか否かを判断し、上記アドレスNが上記アドレスL-1に等しくなければ、再び前述のステップ2の処理に戻るようになっている。

【0047】また、上記アドレスNが上記アドレスL-1に等しいならば、ビット数格納エリアに格納されたビット数Sに+1した数値S+1が、EPROM24のアドレスLに書込まれている1バイトデータの内容(チェックサム値)に一致したか否かを判断する(メモリチェック手段)。すなわち、EPROM24のアドレスLに書込まれたチェックサムデータによりメモリチェックを行う。

【0048】ここで、上記数値S+1が、上記アドレスLのチェックサム値と一致すれば、反転制御回路27の出力をローレベルのままにして、EPROM24から全データを読取る。

【0049】また、上記数値S+1が、上記アドレスLのチェックサム値と等しくなければ、EPROM24には反転データが書込まれていると判断して、反転制御回路27へのデータ線(D0)にハイレベルの信号を出力したのち、IOWR-N出力端子(IOW)から反転制御回路27へ立上がり信号を出力し、反転制御回路27の出力をハイレベルにして、EPROM24から全データを反転して読取る(反転読取手段)。

【0050】上述したEPROM24からの全データの読取りが終了すると、このデータ読取処理を終了するようになっている。

【0051】このような構成の本実施例においては、EPROM24に記憶されたプログラムデータは、EPROM24の各アドレス毎に、すなわち読取る1バイトデータのアドレス毎に、0の状態(データ)のビット数が計数され、EPROM24のアドレスL-1のデータま

8

での0の状態のビット数の合計が算出される。

【0052】さらにこのアドレスL-1までの0の状態の合計ビット数に+1加算して、チェックサムの照合するためのデータが算出される。チェックサムの正值データは、EPROM24のアドレスLに記憶されており、メモリチェックとして、上述した合計ビット数に+1加算したデータ(数値)とEPROM24のアドレスLに記憶されているデータ(数値)とが一致したか否かを判断する。

10 【0053】この判断で一致したと判断された場合には、反転制御回路27の出力をローレベルのままにして、EPROM24からプログラムデータを読取る。

【0054】また、その判断で、一致しないと判断された場合は、すなわちメモリチェックによりエラーと判断された場合には、反転制御回路27の出力をハイレベルにして、EPROM24からプログラムデータを反転して読取る。

【0055】このように本実施例によれば、EPROM24からデータバス25へ接続された8本のデータ線(D0~D7)に介挿されたエクスクルーシブOR回路260~267の各残る一方の入力端子へ共通に信号を出力する反転制御回路27とを設け、チェックサムの結果が正常と判断された場合には、反転制御回路27の出力をローレベルのまま、EPROM24からプログラムデータを読取り、チェックサムの結果がエラーと判断された場合には、反転制御回路27の出力をハイレベルにしてEPROM24からプログラムデータを反転して読取ることにより、前述の第1の実施例のROMライタによりデータを書き込んだ再書き込み可能なROM(EPROM)から、正確にプログラムデータを読取ることができる。従って、EPROM24の寿命を延ばすことができる。

【0056】なお、この実施例においては、フリップフロップ回路からなる反転制御回路27とエクスクルーシブOR回路260~267とを設けたものについて説明したが、この発明はこれに限定されるものではなく、メモリチェックによりエラーと判断されたときに、読取るデータを反転する回路又は反転処理を行うものならば適用できるものである。

40 【0057】また、上述した2つの実施例においては、再書き込み可能なROMとしてEPROM14、24について説明したが、この発明はこれに限定されるものではなく、例えばEEPROM(electrically erasable programmable read only memory)やフラッシュメモリ等についても適用できるものである。

【0058】

【発明の効果】以上詳述したようにこの発明によれば、プログラム等のデータを書込むときに、ビットを1の状態から0の状態に変化させる数を少なくすることができ、従って再書き込み可能なROMの寿命を延ばすことが

できるデータ書込装置及びデータ読取装置を提供できる。

【図面の簡単な説明】

【図1】この発明の第1の実施例のROMライタの要部回路構成を示すブロック図。

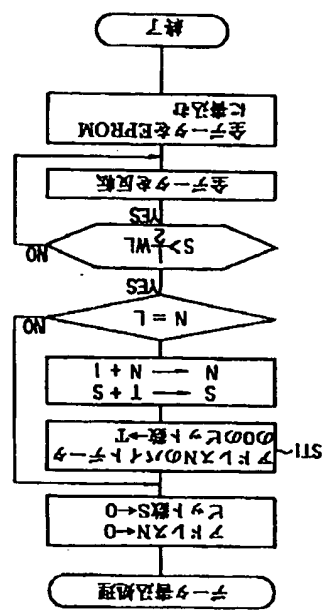
【図2】同実施例を示すデータ書込処理の流れを示すフロー図。

【図3】この発明の第2の実施例のシステムのEPRO Mの読取部の要部回路構成を示すブロック図。

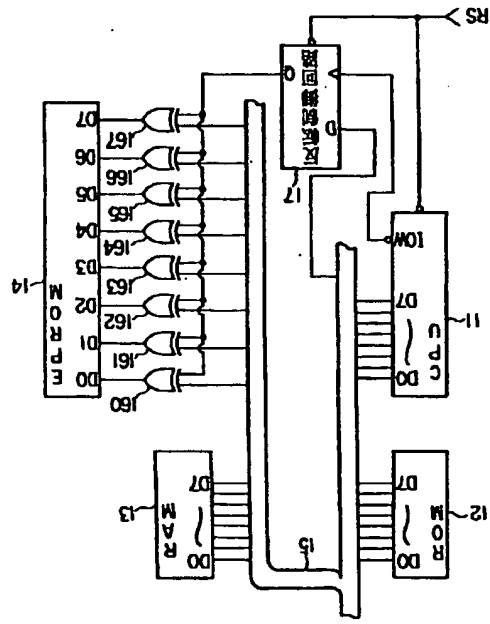
【符号の説明】
11...CPU、14、24...EPROM、15、25...データバス、160~167、260~267...エクスチルーシPOR回路、17、27...反転制御回路。

【図4】同実施例を示すデータ読取処理の流れを示すフロー図。

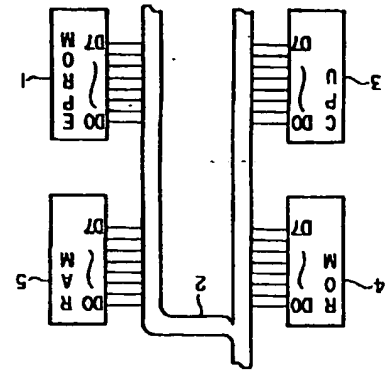
【図2】



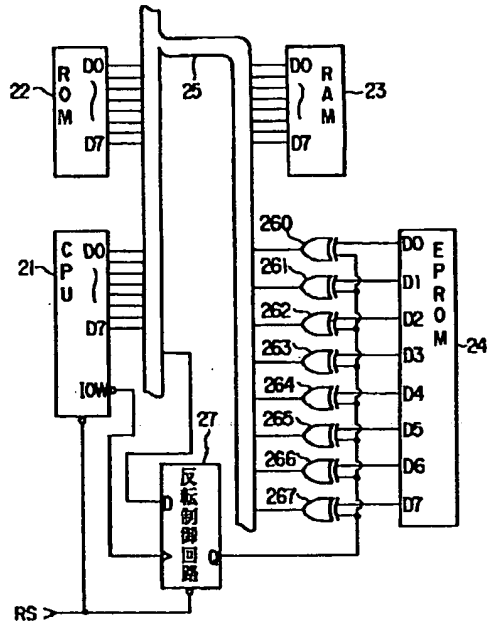
【図1】



【図5】



【図3】



【図4】

